

# Programming the HP 41-CV Calculator

Part 2 of 2 parts

## USER INSTRUCTIONS

After all traverse data has been entered, this program processes the latitudes and departures of the traverse legs stored in the storage registers of the calculator memory.

It calculates the relative accuracy, carries out a Compass or Transit Rule adjustment, and converts the adjusted lats. and depts. to final co-ordinates.

After this conversion, the area of a closed traverse may be computed.

When the program has been loaded into memory, Key A or Key B must be pressed. Key A would be selected for a traverse to a known point other than the starting point. In this event, the display will prompt for the known co-ordinates of the end point to be entered.

When a perfect closure occurs, the relative accuracy will print-out as zero.

Either the Compass Rule or Transit Rule balancing method must be selected, Key C or Key D. Before making this selection, the opportunity may be taken to not balance, or to HOLD, certain legs of the traverse. If it is elected to do this, Key F, and key in the starting point number and end point number in the 000.000 format (for example 5.006 or 62.063) of each traverse leg it is decided to HOLD.

Note that sideshots will be excluded from the adjustment automatically by the calculator. It is not necessary to identify these under this routine.

After the starting and end point numbers of all traverse legs not to be balanced have been keyed in, press key C or key D to carry out the adjustment.

After the adjustment, press Key E if the area within a closed traverse is required.

	Key	Display	Enter
Open-ended Traverse	A	NORTH	Northing of end point
	R/S	EAST	Easting of end point
	R/S	Print-out:-	Error in latitudes
			Error in departures
			Relative accuracy
Closed Traverse	B	Print-out:-	Error in latitudes
			Error in departures
			Relative accuracy
Hold	F	HOLD	Station numbers (000.000)
	R/S	HOLD	etc.

Figure 1

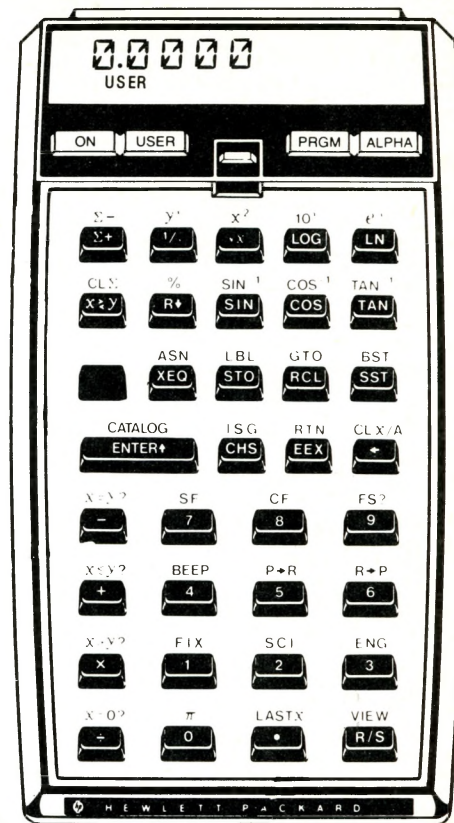
THE FIELD Adjustment program computes the errors in a traverse, its relative accuracy, and processes the latitudes and departures previously calculated and stored (see Field Traverse Program), and converts these to co-ordinates.

The 'User Instructions' are in Figure 1, and a flow chart of the program steps following Labels A and B is in Figure 2.

The known co-ordinates of the end point of an open-ended traverse are accepted and stored in work registers 08 and 09 of the calculator when Key A is pressed.

After Key B, the known co-ordinates of the starting point are recalled from memory and stored in registers 08 and 09.

From line 20 onwards, the program



instructions are the same for both an open-ended traverse and a closed traverse.

At line 20, the error in latitudes (difference between the calculated northing in register 03 and the known northing in register 08) is computed, and because the known northing isn't needed any more, this error is stored in register 08. The error in departures is computed and stored in 09.

The relative accuracy is determined by dividing the main traverse distance from register 02, by the error distance, which is the square root of the sum of the squares of the two errors, of course.

Both errors are squared and summed. In the event of a perfect closure, this sum will be zero. If an attempt is made to divide any number by zero, a computer error will occur, and the program will fail.

A test is made, and if "X = 0?", processing branches around the next few instructions.

Another branch is made if Flag 21 is not set, which indicates that a printer is not attached to the calculator.

### HOLD

The program steps for this function are diagrammed in the lower portion of column 1 in the larger flow chart in Figure 3.

Those who read the description of

the Field Traverse program, will recall that each calculated latitude of a side-shot was added to nine million before being stored in the memory of the calculator, so that they could be identified as such later.

The latitudes to be "Held" will be added to four million.

After Key F has been pressed, and the first pair of station numbers have been keyed in, the integer portion of the number entered is discarded.

Because the latitudes and departures of a traverse leg are stored in registers whose numbers are calculated from the formulas  $2n + 8$  and  $2n + 9$ , where 'n' is the station number that "the traverse leg is going to", the lower number that was keyed in is not required. The only reason for both numbers to be keyed in is to make the User Instructions more precise and unambiguous.

The fractional part of the number is converted to the number of the register containing the latitude of the traverse leg, by multiplying the fraction by two thousand, and adding eight to the product. This short function is set up as a sub-routine, and is used not only here, but several times later whenever a new counter has to be set up.

The latitude is recalled to the "x" register, and tested to see if it is greater than one million. If it is, it means that the user has made a mistake by identifying a sideshot.

Some years ago, when our office was in to 'third order control', and we were setting up fairly large survey networks throughout the city, a professional programmer and myself prepared a program to adjust these networks by the Junction Point method on an I.B.M. 360 main frame computer.

Large amounts of data were to be processed, and special forms were designed for this data to be listed, and subsequently key punched onto dozens, and sometimes even hundreds, of standard computer cards. One key punch operator would punch the card, and a second would verify that it had been done correctly.

Early in the debugging process, we found that incorrect data was going into the computer. The first operator was keying data into columns one and two of a card instead of into columns zero and one, and the second operator was making the same mistake.

Instead of shouting at the girls, and getting them into trouble, etc. etc, the programmer merely changed the program so that this data was supposed to be in columns one and two on the punch-

ed card, and we had no further trouble with this aspect of data entry.

I have thought about this small episode from time to time, and sometimes feel that in this minor conflict the machine won, and that the human race, in whatever a small part, was somehow degraded.

Something similar occurs in this program.

Although the 'User Instructions' say that sideshots will be with-held from the adjustment automatically, and need not be identified in this routine, you just know that someone, sometime, will forget, and will key in a sideshot. If this happens, the latitude plus the nine million already in storage will have another four million added to it, and the whole calculation will eventually be ruined.

The test, therefore, is to catch such a mistake, and to not add four million to the nine million already in place.

If the latitude is less than one million, four million is added to it, and the latitude is stored back in memory.

Processing goes back to Label F to accept the next pair of numbers, if any, that identify a traverse leg to be 'Held'.

### CONSTANTS

As is well known, in a Transit Rule adjustment, the correction to a latitude (or departure) is equal to the latitude (or departure) multiplied by "the total error in latitudes (or departures) divided by the total of all the latitudes (or departures)".

The Compass Rule correction is equal to the distance of the traverse leg multiplied by "the total error in latitudes (or departures) divided by the total distance".

The first phase of the adjustment computation is to calculate the constants included within the quotation marks in the above statements.

The phrases 'total of all the latitudes (or departures)' and 'total distance' are not strictly true, because that data pertaining to sideshots, and those traverse legs not to be adjusted (HOLD) must be excluded from these totals.

Flag 00 is set immediately after Label C at step 070 as an indication that a Compass Rule adjustment is to be carried out. If Key D is pressed, Flag 00 is not set, and a Transit Rule adjustment will take place.

The sub-routine to calculate a new counter from the fractional portion of the number in register 05 (see Field Traverse program, described earlier) is called, and the new counter is stored in register 04.

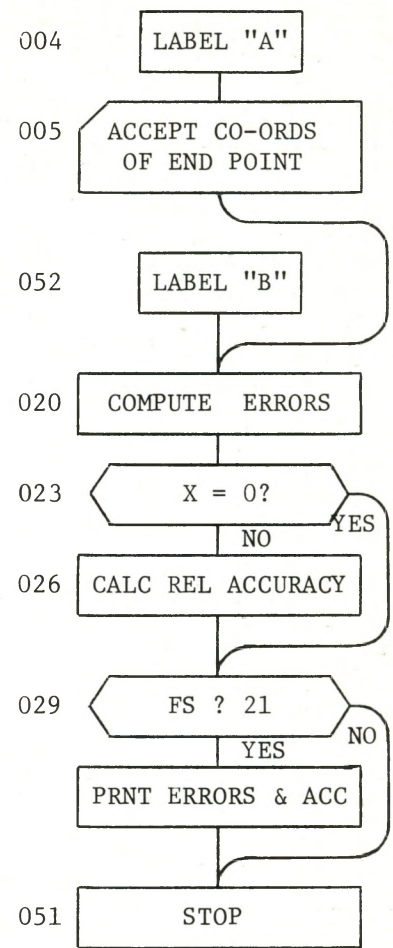


FIGURE 2

Registers 02, 03, 06 and 07 are cleared of data by storing zeros therein.

The counter is incremented at line 079, and is incremented again at line 083.

The new counter is recalled from register 04 to the Y register, and the old counter is recalled from register 05 to X. The test "Is Y greater than X?" is made. When the answer is positive it means that all the latitudes and departures or all the distances have been added, that this phase of the calculation is complete, and that the program should branch to line 120.

When the new counter is still less than the old, processing continues to recall the next latitude from the storage register indicated by the new counter.

This latitude is tested, and if it is greater than one million, indicating either a sideshot or a 'Hold', then a branch is made back to line 079 to increment the counter two times.

If the latitude is less than one million, Flag 00 is tested. If it is not set (Transit Rule), the latitude is given a plus sign (absolute value), and added into register 02. The counter is incremented, the departure is recalled from memory, ab-

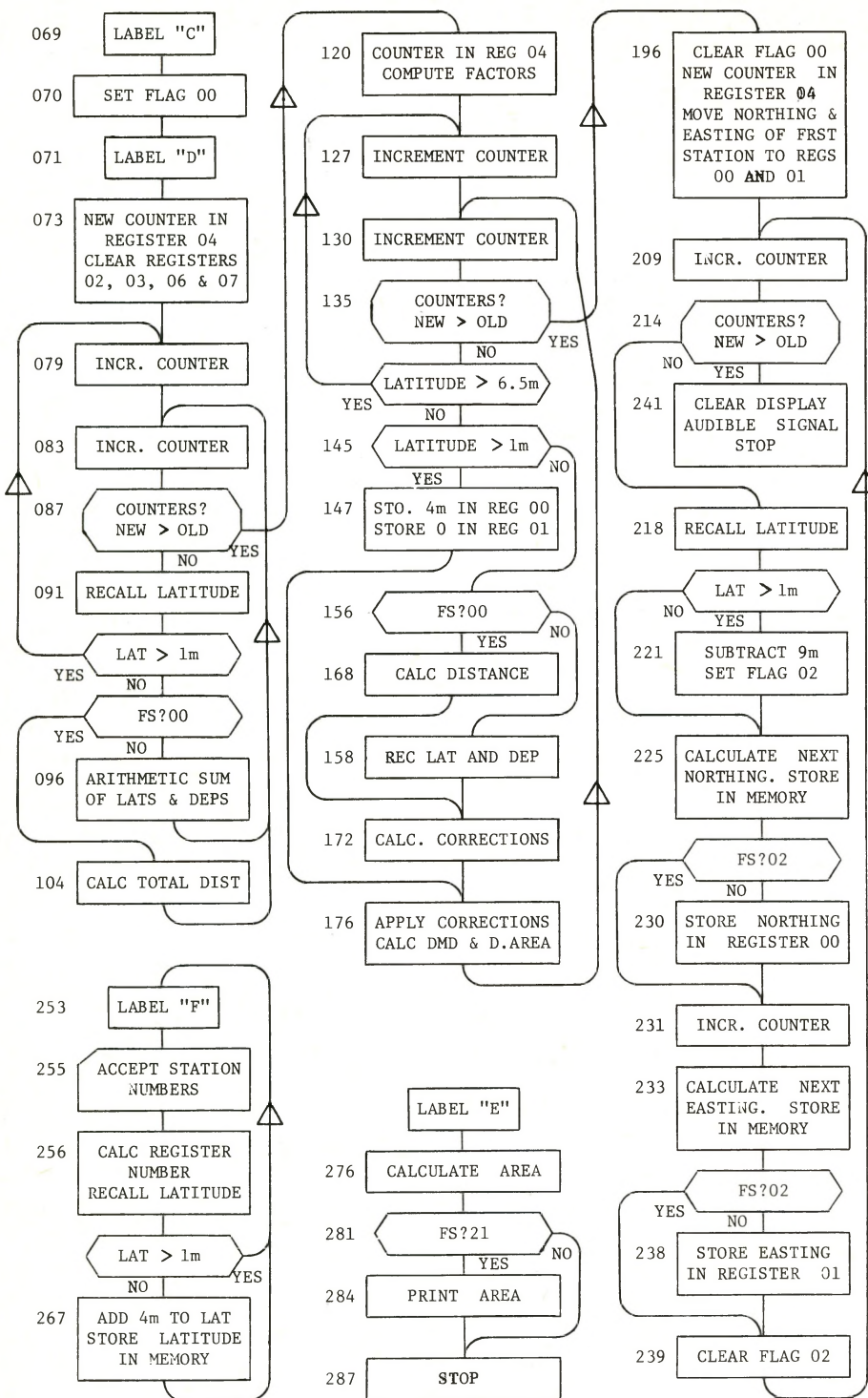


FIGURE 3

solute value, and it is added into register 03. Processing loops back to line 083 to increment the counter one time.

If Flag 00 is set (Compass Rule), a sub-routine is called that will calculate the traverse leg distance. In this sub-routine, the latitude is squared, the counter is incremented, the departure is re-

called, and it is squared. The two squares are added, and the square root of the sum is extracted.

The calculated traverse leg distance is added into both register 02 and register 03, and processing loops back to line 083 to increment the counter one time.

When processing branches from line 087 to line 120, the contents of register 02 are recalled and divided into the total error in latitudes in register 08. The contents of register 03 are recalled and divided into register 09. The two constants required are now in registers 08 and 09.

### CORRECTIONS

The second phase of the adjustment which begins at line 127 is to compute and apply each correction, and at the same time carry forward a double meridian distance, and calculate a double area.

A branch, not shown in the flow chart, is made to the sub-routine that will set up a new counter in register 04.

The new counter is incremented at lines 127 and 130, and it is tested with the old counter. When the value of the new counter is greater than the old, this second phase is complete, and a branch is made to line 196.

At line 141, the latitude in the register indicated by the counter is recalled and tested. If it is greater than 6.5 million it is a latitude of a sideshot and processing loops back to line 127 to increment the counter two times.

The latitude is tested again, and if it is less than one million, a branch is made to line 156 where Flag 00 is tested.

If it is set, and a Compass Rule adjustment is in progress, the sub-routine already mentioned to calculate the traverse leg distance is called. The calculated distance is stored in registers 00 and 01.

The factor from register 08 is recalled and multiplied into register 00 to compute the correction to the latitude.

The factor from register 09 is recalled and multiplied into register 01 to compute the correction to the departure.

During the sub-routine to calculate the traverse leg distance, the counter was incremented by one.

At line 177 the departure indicated by the counter is recalled, and the correction in register 01 is subtracted from it.

The corrected departure is added into register 07 to compute the double meridian distance, and the corrected departure is also stored back in the main memory.

The counter is decremented by one.

The latitude is recalled, its correction from register 00 is subtracted from it, and the corrected latitude is stored back in the main memory.

At this time the corrected latitude is multiplied by the D.M.D. which is recalled from register 07; the product is added into register 06 where the double area is being accumulated.

The counter is incremented, and the corrected departure is recalled from the main memory and added into register 07 to up-date the D.M.D.

Processing returns to line 130 to increment the counter one time.

If Flag 00 had not been set at line 156, the latitude would have been stored in register 00, the counter incremented, and the departure recalled and stored in register 01.

The corrections, D.M.D. and double area would have been computed by recalling the factors from 08 and 09, etc. as described above.

If the test at line 145 had indicated a latitude in excess of one million, then it would be a latitude that is not to be adjusted, but which does form part of the main traverse, and which must play its part in the computation of the area.

Because this is the last time that this latitude needs to be identified as a 'Hold', the opportunity is taken to put four million into register 00 and a zero into register 01. Then processing is sent to line 177 where these two values are treated as corrections by the program instructions, and are subtracted from the latitude and departure. The unadjusted lat. and dep. are used to up-date the D.M.D. and the double area as described.

### CO-ORDINATES

The final phase of the adjustment process converts the adjusted latitudes and departures into pairs of co-ordinates.

The sub-routine to set up a new counter in register 04 is called, and the new counter is first used to recall the northing and easting of the first station from the memory to work registers 00 and 01.

The counter is incremented once, and tested. When it exceeds the old counter, all of the conversions have been made, and the adjustment is complete. The display is cleared for neatness, and because the routine may have taken several minutes to run, a tone is sounded as an audible signal to the user, before the program stops at line 243.

Until this occurs, the latitude is recalled from the storage register indicated by the counter.

It is tested to see if it is a sideshot, and if it is, nine million is subtracted from it, and Flag 02 is set.

The northing is recalled from work register 00 and added to the latitude to

calculate the northing of the next station. This new northing is immediately stored in the storage register indicated by the counter, thus erasing the latitude lying therein.

If Flag 02 is not set, the new northing is stored back in register 00 so that the next latitude to be recalled can be added to it in due course.

If Flag 02 is set, however, then the new northing of the sideshot is not stored in register 00; because it is not on the main traverse, it will not be required to compute the next northing.

The counter is incremented, and in the same way the next easting is calculated, stored back in memory in place of the departure, and then stored in register 01 only if Flag 02 is not set.

The instruction "Clear Flag 02" is made before the program loops back to line 209 to increment the counter one time.

When a closed traverse has been processed, Key E is pressed if an area is required. The double area is recalled from register 06, divided by two, truncated to one decimal place for appearance, and either printed if a printer is attached, or else left in the display.

### ROTATING SIDESHOTS

Programs such as these can always be improved, by either making them do more things, or else by making them do the same thing but using fewer instructions, thus making them more efficient.

Discerning readers will have noticed a shortcoming in the above program in that sideshots do not reflect any rotation in azimuth that may occur due to the adjustment to the adjacent main traverse leg.

For example, if the latitude and departure of traverse leg 61 - 62 (see Figure 3 in Field Traverse program description) were adjusted to change the bearing of this line by say six seconds of arc, this would not in any way change the lat. and dep. of sideshot 62 - 63, and consequently the field angle of 39° 08' to the sideshot would not be accurately represented by the computed co-ordinates.

A solution to this is available, and for completeness it is shown in the flow chart in Figure 4. It requires sixty-two lines of program instructions, which occupy an additional fourteen storage registers in the memory of the calculator. Fourteen registers that could otherwise contain pairs of co-ordinates on seven survey stations.

The problem with these relatively small calculators, is that compromises such as these frequently have to be made.

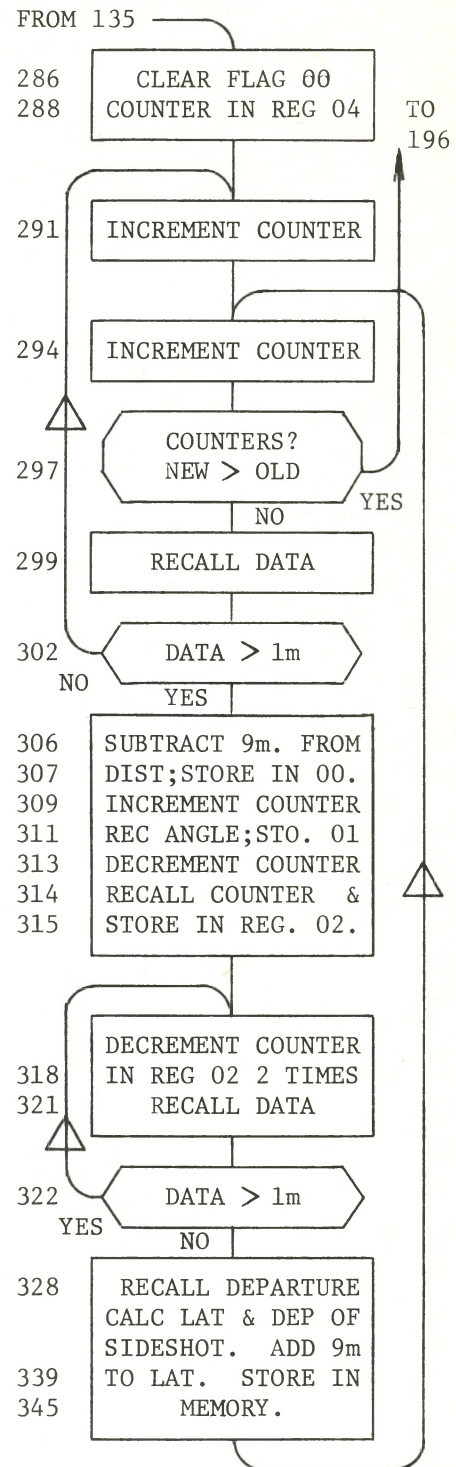


FIGURE 4

Fine touches to the program have to be left out, and less than the best results accepted, in trade for more storage for computed data.

The amendment to this program requires that in the Field Traverse program the distance to the sideshot be added to nine million and stored in place of the latitude, and that the angle that the side-

cont'd on page 14

shot makes with the preceding main traverse leg be stored in place of the departure.

The chart in Figure 4 would fit between the tops of columns two and three in Figure 3, and is mostly self explanatory.

If there are no sideshots along the main traverse, processing will loop between lines 291 and 302 until all the tests have been made.

When a sideshot is encountered, the distance and angle are recalled from memory to registers 00 and 01.

The instructions in lines 318 to 328 locate the latitude and departure of the adjacent main traverse leg. The loop at 322 is required because there may be more than one sideshot at one main traverse station.

The back azimuth of the main traverse leg is computed from the adjusted latitude and departure with the 'rectangular to polar' function, the angle in register 01 is added to it, and then with the computed forward azimuth and the sideshot distance, the 'polar to rectangular' function will calculate a latitude and departure of the sideshot that will accurately reflect the field work. ●